WolfSSL

Luca Valentini

Insert Date

Contents

Abstract

Explanation of this article. Must be a synthesis

Chapter 1

SSL Protocol

1.1 Introduction

The SSL protool is a client/server protocol that provides the following basic security services to the communicating peers:

- Authentication (both peer entity anda data origin authentication) services
- Connection confidentiality services
- Connection integrity services

The SSL protocol is sockets-oriented, meaning that all or none of the data that is sent to or received from a network connection is cryptographically protected in exactly the same way. It can be best viewed as an intermediate layer between the transporrt and the application layer that serves two purposes:

- Establish a secure connection between the commucating peers
- Use this connection to securely trasmit giher-layer protocol data from the sender to the reciever. It therefore fragments the data in pieces called fragments; each fragment is optionally compressed, authenticated, encrypted, prepended with a header, and transmitted to the reciever. Each data fragment prepared this way is sent in a distinct SSL record.

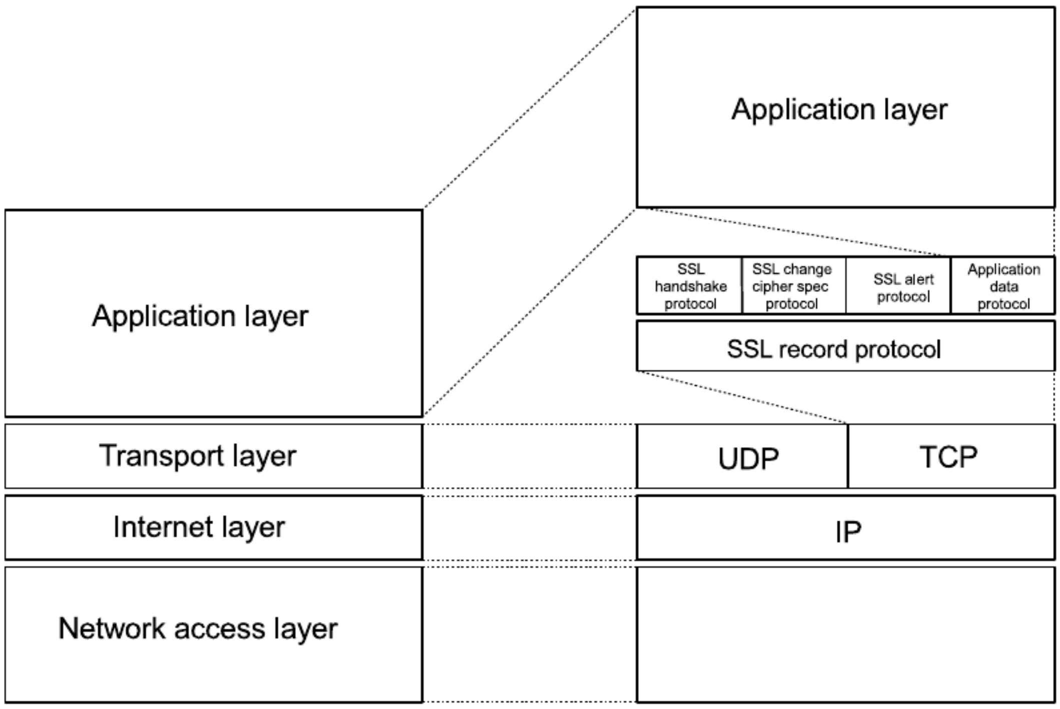

Figure 1.1: The SSL with its (sub)layer and (sub)protocols

The SSL consists of two sublayers and a few subprotocols:

- The lower sublayer is stacked on top of some connection-oriented and reliable transport layer protocol. This layer basically comprises the SSL record protocol that is used for the encapsulation of the higher-layer protocol data.
- The higher sublayer is stacked on top of the SSL record protocol and comprises four subprotocols.
  - The *SSL handshake protocol* is the core subprotocol of SSL. It is used for establishment of a secure connection. It allows the communicating peers to authenticate each other and to negotiate a cipher suite and a compression method.
  - The *SSL change cipher spec protocol* is used to put the parameters, set by the SSL handshake protocol in place and make them effective.
  - The *SSL alert protocol* allows the communicating peers to signal indicators of potential problems and send respective alert messages to each other.

– The *SSL application data protocol* is used for the secure transmission of application data.

In spite of the fact that SSL consists of several subprotocols, we use the term *SSL protocol* to refer to all of them simultaneously.

1.2 Valutare se aggiungere qualcosa su SSL. Non vorrei andare fuori tema

***********Valutare se aggiungere qualcosa su SSL. Non vorrei andare fuori tema ***********

Chapter 2

Mettere l'handshake nel capitolo 1

Chapter 3

Wolf SSL

The wolfSSL embedded SSL library is a lightweight SSL/TLS library written in ANSI C and targeted for embedded, RTOS, and resource-constrained environments - primarily because of its small size, speed, and feature set.
It's free and it has an excellent cross platform support.
WolfSSL supports standards up to the current TLS 1.3 and DTLS 1.2 levels, is up to 20 times smaller than OpenSSL and it's powered by the colfCrypt library.

This library is built for maximum portability and supports the C programming language as a primary interface. It also supports several other host languages, including Java (wolfSSL JNI), C# (wolfSSL C#), Python, and PHP and Perl.

To improve performance it supports hardware cryptography and acceleration on several platforms.

In the following list you can see some of WolfSSI's features:

- Runtime memory usage between 1-36 kB
- OpenSSl compatibility layer
- Hash Functions:
  - MD2
  - MD4
  - MD5
  - SHA-1
  - SHA-224
  - SHA-256
  - SHA-384
  - SHA-512
  - BLAKE2b
  - RIPEMD-160
  - Poly1305
- Mutual authentication support (client/server)

- SSL Sniffer (SSL Inspection) Support
- IPv4 and IPv6 support

The operating systems supported are:

1. Win32/64
2. Linux
3. Mac OS X
4. Solaris
5. ThreadX
6. VxWorks
7. FreeBSD
8. NetBSD
9. OpenBSD
10. embedded Linux
11. Yocto Linux
12. OpenEmbedded
13. WinCE
14. Haiku
15. OpenWRT
16. iPhone(iOS)
17. Android
18. Nintendo Wii and Gamecube through DevKitPro
19. QNX
20. MontaVista
21. NonStop
22. TRON / ITRON / ITRON
23. Micrium C / OS - III
24. FreeRTOS
25. SafeRTOS
26. NXP / Freescale MQX
27. Nucleus
28. TinyOS
29. HP / UX
30. AIX
31. ARC MQX
32. TI - RTOS
33. uTasker
34. embOS
35. INtime
36. Mbed
37. uT - Kernel
38. RIOT
39. CMSIS -RTOS
40. FROSTED
41. Green Hills INTEGRITY
42. Keil RTX
43. TOPPERS
44. PetaLinux
45. Apache Mynewt
46. PikeOS